A SPOKEN LANGUAGE SYSTEM FOR AUTOMATED CALL ROUTING

G. Riccardi A. L. Gorin A. Ljolje M. Riley

AT&T Labs-Research
600 Mountain Avenue
Murray Hill, NJ 07974, USA
{dsp3,algor,alj,riley}@research.att.com

1. ABSTRACT

We are interested in the problem of understanding fluently spoken language. In particular, we consider people's responses to the open-ended prompt of *'How May I help you?'*. We then further restrict the problem to classifying and automatically routing such a call, based on the meaning of the user's response. Thus, we aim at extracting a relatively small number of semantic actions from the utterances of a *very large set* of users who are *not trained* to the system's capabilities and limitations. In this paper, we describe the main components of our speech understanding system: the large vocabulary recognizer and the language understanding module performing the call-type classification. In particular, we propose automatic algorithms for selecting phrases from a training corpus in order to enhance the prediction power of the standard word n-gram The phrase language models are integrated into stochastic finite state machines which outperform standard word n-gram language models. From the speech recognizer output we recognize and exploit automatically-acquired salient phrase fragments to make a call-type classification. This system is evaluated on a database of $10K$ fluently spoken utterances collected from interactions between users and human agents.

2. INTRODUCTION

The typical approaches to the problem of topic classification are word and concept spotting. Although these techniques work quite well for small applications, they do not scale up to large tasks and are limited in scope. On the other hand, we view this problem as understanding speech by taking into account the information conveyed by the whole utterance, with the ultimate goal of building automatically trained language models integrating both recognition and understanding. For this reason, we use a large vocabulary speech recognition front-end followed by an understanding module which performs a stochastic mapping between *salient fragments* and call-types.

The problem of automated call routing has been addressed in [2] and the issues concerning the understanding and dialog mechanisms are part of ongoing research ([1], [10]). We have created a database of 10K spoken transactions of people responding to a human agent's greeting of *'How May I help you?'* [2]. The first utterance of each transaction has been transcribed and marked with a call-type by labelers. There are 14 call-types plus an *other* class as a complement. In particular, we focused our study on the classification of the user's first utterance in these dialogs. The spoken sentences vary widely in duration, with a distribution distinctively skewed around a mean value of 5.3 seconds corresponding to 19 words per utterance. Some examples of these utterances are given below:

- `Yes ma'am where is area code two zero one?`
- `I'm tryn'a call and I can't get it to go through I wondered if you could try it for me please?`
- `Hello`

The whole set of utterances has been split into three subsets for training (8K), developing (1K) and testing (1K) the acoustic and language models for recognition and understanding. In the the training set there are $3.6K$ words which define the lexicon. The out-of-vocabulary (OOV) rate at the token level is 1.6%, yielding a sentence-level OOV rate of 30%. Significantly, only 50 out of the 100 lowest rank singletons were cities and names while the other were regular words like *authorized*, *realized*, etc.

3. LANGUAGE MODELING

For language modeling, to constrain the recognizer, we automatically trained stochastic grammars represented with the Variable Ngram Stochastic Automaton (VNSA) [5]. The VNSA is a non-deterministic automaton that allows for parsing any possible sequence of words drawn from a given vocabulary V. Moreover, it implements a backoff mechanism to compute the probability of unseen word-tuples. The stochastic automaton is automatically generated from the training corpus according to the algorithm presented in [5]. The *order* of the VNSA network is the maximum number of words that can be used as left context in the computation of the conditional probability $P(w_i|w_{i-n+1},\ldots,w_{i-1})$. VNSAs have been used to approximate standard n-gram language models and their performance is similar to the standard bigram and trigram models [5]. The benefit from the use of the VNSAs is threefold. First, the incorporation into a one-pass Viterbi speech decoder is straightforward and efficient. Second, VNSAs can be exploited in a cascade of transducer compositions for speech processing (e.g., to include intra and interword phonotactic constraints) [8]. Thirdly, the VNSA is an effective method for training and implementing stochastic class-based language models that outperform the standard n-gram models in terms of perplexity and word accuracy [4], [5].

4. LANGUAGE MODELING WITH AUTOMATICALLY ACQUIRED PHRASES

Traditionally, standard n-gram language models for speech recognition implicitly assume *words* as the basic lexical unit. However, the motivation for choosing optimal longer units for language modeling is threefold. First, not all languages have a predefined unity, such as the *word* (e.g. the chinese language). Second, many word tuples (phrases) are recurrent in the language and can be thought as a single lexical entry (e.g. `by and large`, `I would like to`, `United States of America`, etc..). Third, the conditional probability $P(w_i|w_{i-n+1},\ldots,w_{i-1})$ can benefit greatly by using variable length units to capture long spanning dependencies, for any given order n of the model. In a previous work, we have shown the effectiveness of incorporating manually selected phrases, into the VNSAs for reducing the test set perplexity and the word error rate of a large vocabulary recognizer([4], [5]). However, a critical issue for the design of a language model based on phrases is the algorithm the automatically chooses the units by optimizing a suitable cost function. For improving the prediction of word probabilities, the criterion we used is the minimization of the language perplexity $PP(\mathcal{T})$ on a training corpus $\mathcal{T}$. This algorithm for extracting phrases from a training corpus is similar in spirit to [6], but differs in the language model components and optimization parameters. In addition, we extensively evaluate the effectiveness of phrase n-gram ($n \geq 2$) language models by means of an end-to-end evaluation of a spoken language system. The phrase acquisition method is a greedy algorithm that performs local optimization based on an iterative process which converges to a local minimum of $PP(\mathcal{T})$. As depicted in fig. 1, the algorithm consists of three main parts:

- Generation and ranking of a set of candidate phrases. This step is repeated at each iteration of algorithm to constrain the search for all possible symbol sequences observed in the training corpus.
- Each candidate phrase is evaluated in terms of the training set perplexity.
- At the end of the iteration, the set of selected phrases is used to filter the training corpus and replace each occurrence of the phrase with a new lexical unit. The filtered training corpus will be referenced as $\mathcal{T}_f$.

In the first step of the procedure, a set of candidate phrases (unit pairs) [1] is drawn out of a training corpus $\mathcal{T}$ and ranked according to a correlation coefficient. The most used measure for the interdependence of two events is the mutual information $MI(x,y) = log\frac{P(x,y)}{P(x)P(y)}$. However, in this experiment, we use a correlation coefficient that has provided the best convergence speed for the optimization procedure:

$$\rho_{x,y} = \frac{P(x,y)}{P(x)+P(y)} \tag{1}$$

where $P(x)$ is the probability of symbol x . The coefficient $\rho_{x,y}$ $(0 \leq \rho_{x,y} \leq 0.5)$ is easily extended to define $\rho_{x_1,x_2,\ldots,x_n}$ for the n-tuple $(x_1,x_2,\ldots,x_n)$ $(0 \leq \rho_{x_1,x_2,\ldots x_n} \leq 1/n)$. Phrases (x,y) with high $\rho_{x,y}$ or $MI(x,y)$ are such that $P(x,y) \simeq P(x) \simeq P(y)$. In the case of $P(x,y) = P(x) = P(y)$, $\rho_{x,y} = 0.5$ while $MI = -logP(x)$. Namely, the ranking by MI is biased towards events with low probability events which are not likely to be selected by our Maximum Likelihood algorithm. In fact, the phrase (x,y) will be selected only if $P(x,y) \simeq P(x) \simeq P(y)$ *and* the training set perplexity is decreased when (x,y) is treated as a single unit. In fig. 2 we show the behavior of the training set perplexity by incorporating an increasing number of selected phrases using $\rho_{x,y}$ and $MI(x,y)$ as ranking coefficient. In particular, after evaluating 1000 phrases and selecting 300 of those, the perplexity decrease is 20% and 4% using $\rho_{x,y}$ and $MI(x,y)$ respectively.

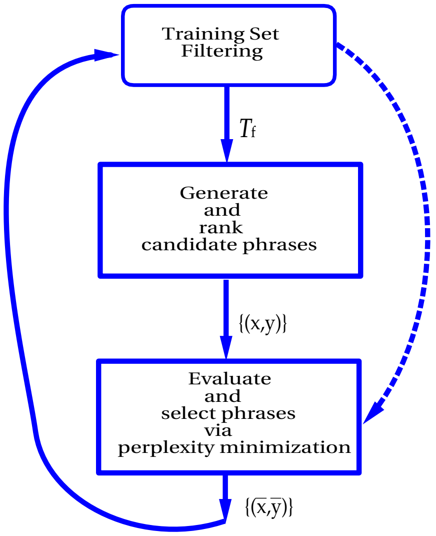
Training Set Filtering
$\mathcal{T}_f$
Generate and rank candidate phrases
{(x,y)}
Evaluate and select phrases via perplexity minimization
$\{(\bar{x},\bar{y})\}$

Figure 1: Algorithm for phrase selection

[1] We ranked symbol pairs and increased the phrase length by successive iteration. An additional speed up to the algorithm could be gained by ranking symbol k-tuples ($k \geq 2$) at each iteration.

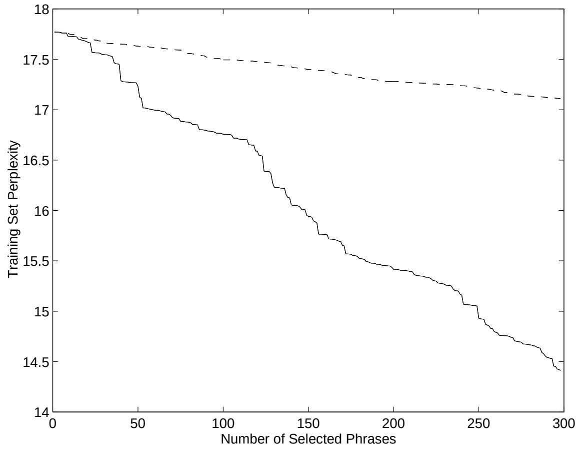

Figure 2: Training set perplexity *vs* number of selected phrases using ρ (solid line) and MI (dashed line).

Each of the candidate phrases (x, y) is treated as a single unit in order to build a VNSA model λ of $k - th$ $(k \geq 2)$ order based on the filtered training corpus, $\mathcal{T}_f$ [2]. Then, (x, y) is selected by the algorithm if $PP(\mathcal{T}) > \mathcal{PP}_\lambda(\mathcal{T})$. The stochastic finite state machine automaton has now the phrase x_y in the set of recognizable symbols. However, the perplexity $PP_\lambda(\mathcal{T})$ is computed at the word level. At the end of each iteration the set $\{(\bar{x}, \bar{y})\}$ is selected and employed to filter the training corpus. The algorithm iterates until the perplexity decrease saturates or a specified number of phrases have been selected. The second issue in building a language model with phrase is the training of the VNSAs with the newly selected lexical units. The algorithm just described provides a segmentation of the training corpus sentences into variable length lexical units. In general, the replacement of the symbol sequence (x, y) with a phrase unit x_y may disable the parsing of symbol sequence of the type $\ldots, z, y, \ldots$ or $\ldots, x, z, \ldots$. Thus, in order to take advantage of the prediction power of the selected phrases without losing the *granularity* of the basic unit lexicon we have considered a multiple parsing strategy. The finite state machine learning algorithm in [4] is provided with all available segmentation of each sentence of $\mathcal{T}$ including that one using *only* basic lexical units. As a result, the VNSA is designed in such a way that it can parse any possible sequence of basic unit while computing state transition probabilities based on the selected phrases ([5]). In addition, the VNSAs' states are pruned in order to minimize the number of states with a negligible increase of the test set perplexity [5].

In figure 3 the test set perplexity is measured versus the VNSA orders for word and phrase language models. It is worth noticing that the highest perplexity payoff comes from using phrase bigram with respect to word bigram. Furthermore, the perplexity of the phrase models is alway lower than the word models, as a result of the multiple segmentation in the training procedure.

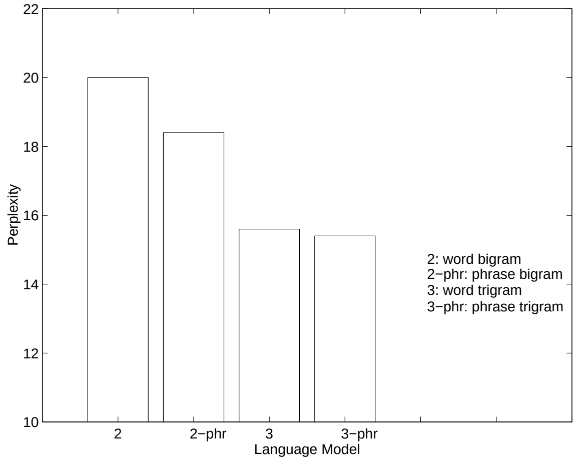

Figure 3: Test set perplexity *vs* VNSA Language Model Order

5. SPEECH RECOGNITION EXPERIMENTS

In these experiments, we used off-the-shelf acoustic models trained on a separate database of telephone-quality read-speech utterances. The lexicon contains a single pronunciation per word which is then composed with the VNSA's stochastic network into a weighted rational transducer. This result is then composed on-the-fly with yet another transducer to apply full-context acoustic phone models [9]. The engine used for speech decoding is a research version of the AT&T Watson recognizer [7]. In table 1 we report the results for word accuracy versus variable VNSA model-order. From these preliminary experiments, we observe a significant advantage only using phrase bigram over word bigram and no payoff from high order VNSAs.

unit type	VNSA order	
	2	3
word	49.5	52.7
phrase	50.5	52.7

Table 1: Word accuracy versus variable VNSA order using words and phrases.

6. CALL-TYPE CLASSIFICATION

For call type classification, we automatically acquired *salient phrase fragments* from the training set of sentence-action pairs [1]. The *salient* phrases have the important property of modeling local constraints of the language while carrying most of the semantic interpretation of the whole utterance. We performed a transduction from observed phrase fragments in an utterance to associated call-types. Then, we used a peak-of-fragments classifier to determine

[2] At the first iteration $\mathcal{T}$ $\equiv \mathcal{T}_f$.

the $1st$ and $2nd$ most likely call-types of the whole utterance. In an automated call router there are two important performance measures. The first is the probability of false rejection, where a call is falsely rejected or classified as *other*. Since such calls would be transferred to a human agent, this corresponds to a missed opportunity for automation. The second measure is the probability of correct classification. Errors in this dimension lead to misinterpretations that must be resolved by a dialog manager [10]. In fig. 4, we plot the probability of correct classification versus the probability of false rejection, for word and phrase language models of increasing order. The curves are generated by varying a salience threshold [3], [1]. Phrase bigrams outperform both word bigrams and trigrams with memory and speed similar to word bigrams. In particular, if we pick the operating point where the false rejection rate is 40%, phrase bigrams ouperform word trigram and we gain 6% accuracy in call-type classification over a word bigrams. We thus conclude that building stochastic local grammars for language modeling and interpretation is a critical research issue for this type of task. In a dialog system, it would be useful even if the correct call-type was one of the top 2 choices of the decision rule [10]. Thus, in fig. 5 the classification scores are shown for the first and second ranked call-types identified by the understanding algorithm.

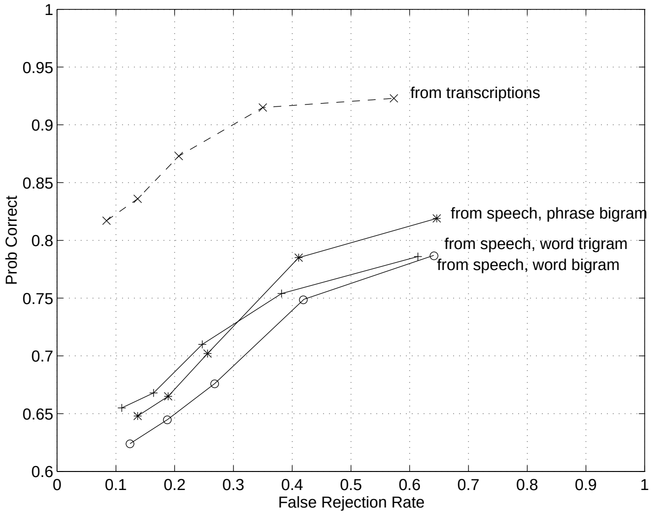

Figure 4: Rank 1 classification rate from text and speech with word n-grams and phrase bigrams

7. CONCLUSION

In conclusion, we have described our preliminary research results on the task of automated call routing for a database of $10K$ utterances. We have described the database and the system we used for large vocabulary speech recognition. Then we proposed methods for automaticallly selecting phrases from a training corpus for both recognition and understanding. Phrase bigrams achieve performances in between word bigram and trigram, in terms of perplexity and word accuracy. In terms of call-type classification, phrase bigrams outperform word trigrams with memory and speed similar to word bigrams. Finally, we have reported on the understanding module and the related call-type classification scores.

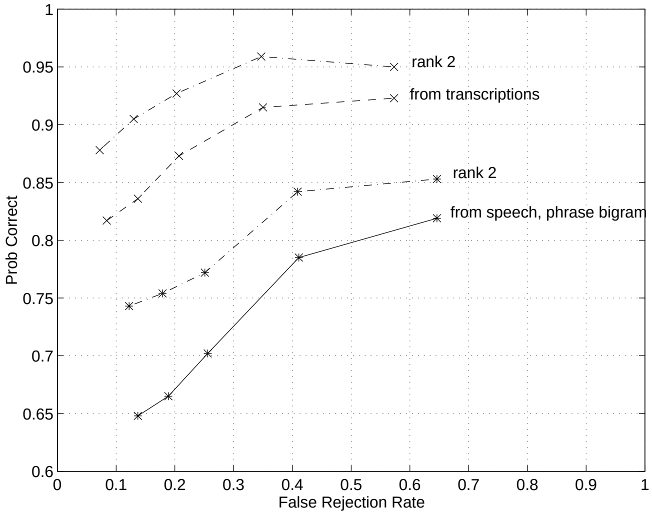

Figure 5: Rank 1 and 2 classification rate plot from text and speech with phrase bigrams

REFERENCES

[1] A.L. Gorin, "Processing of semantic information in fluently spoken language" *Proc. ICSLP*, pp. 1001-1004, Philadelphia, 1996.

[2] A.L. Gorin, B.A. Parker, R.M. Sachs and J. G. Wilpon, "How may I help you?" *Proc. IVTTA*, pp. 57-61, Basking Ridge, 1996.

[3] A.L. Gorin, "On automated language acquisition," *J. Acoust. Soc. Am.*, 97, pp. 3441-3461, 1995 .

[4] G. Riccardi, R. Pieraccini and E. Bocchieri, "Non-Deterministic Stochastic Language Modeling for Speech Recognition ," *Proc ICASSP*, pp. 237-240, Detroit, 1995.

[5] G. Riccardi, R. Pieraccini and E. Bocchieri, "Stochastic automata for language modeling ," *to appear in Computer Speech and Language*, 1997.

[6] E. Giachin, "Phrase Bigrams for Continuous Speech Recognition" *Proc ICASSP*, pp. 225-228, Detroit, 1995.

[7] R. D. Sharp et al., "The WATSON Speech Recognition Engine" *Proc. ICASSP*, Munich, 1997.

[8] F. Pereira, M. Riley and R. Sproat, "Weighted rational transduction and their application to human language processing," *Proc. Workshop on Human Language Technology*, pp. 249-254,Austin, 1994.

[9] M. Riley, F. Pereira and E. Chung, "Lazy transducer composition: a flexible method for on-the-fly expansion of context-dependent grammar networks,"*Proc. ASR Workshop*, Snowbird, 1995.

[10] S. Boyce and A.L. Gorin, "User interface issues for natural spoken dialog systems,"*Proc. ICSLP*, pp. 1577-1580, Philadelphia, 1996.